



UNICEUB – CENTRO UNIVERSITÁRIO DE BRASÍLIA
FACULDADE DE CIÊNCIA EXATAS E TECNOLOGIA –
FAET
RICARDO KOITI ISHIDA

**SIMULAÇÃO DE SISTEMA DE AUTOMAÇÃO RESIDENCIAL PARA DEFICIENTES
COM DIFICULDADES MOTORAS**

Brasília – Distrito Federal
07/2004



UNICEUB – CENTRO UNIVERSITÁRIO DE BRASÍLIA
FACULDADE DE CIÊNCIA EXATAS E TECNOLOGIA –
FAET
ENGENHARIA DA COMPUTAÇÃO
RICARDO KOITI ISHIDA 9966383

SIMULAÇÃO DE SISTEMA DE AUTOMAÇÃO RESIDENCIAL PARA DEFICIENTES COM DIFICULDADES MOTORAS

Projeto Final do curso de Engenharia da Computação

Prof.[*msc*] ***Maria Marony Sousa Farias Nascimento***
Orientadora

Brasília/Distrito Federal,
2004.

Dedicatória

À minha família e amigos, pelo incentivo e apoio que me deram para a conclusão deste projeto.

Agradecimentos

Agradecimentos especiais ao engenheiro, professor doutor e coordenador do curso de Graduação de Engenharia da Computação Abiezer A. Fernandes e aos engenheiros e professores Maria Marony Sousa Farias Nascimento e Francisco J. De Obaldia pela valiosa ajuda e contribuição que me prestaram, fundamentais para a realização do projeto e elaboração desta monografia.

Resumo

Dada a dificuldade de locomoção dos deficientes paraplégicos, quanto mais automatizadas as tarefas, maior conforto e maior comodidade poderão ser fornecidos a eles.

Tarefas consideradas simples para uma pessoa dotada de capacidade física de locomoção, como verificar se as portas e janelas de sua própria residência estão abertas ou fechadas e executar o ato de abrir e fechar, tornam-se complexas para um deficiente paraplégico.

Em função das dificuldades encontradas pelos deficientes paraplégicos na locomoção em sua própria residência e com o objetivo de reduzir o esforço físico de modo a facilitar seu dia-a-dia, este projeto apresenta uma solução que permitirá ao usuário, sem necessidade de se locomover, ter conhecimento do estado de sua residência, ou seja, verificar se as portas e as janelas estão abertas ou fechadas e executar o ato de abrir ou fechá-las.

Este projeto visa apresentar uma simulação da maneira pela qual o deficiente poderá ter uma visão geral do estado de cada parte da casa automatizada e enviar ordem sem necessidade de locomoção.

Com essa ferramenta, o usuário poderá verificar, por si mesmo, sem esforço físico, o estado de sua residência e enviar uma determinada ordem ao periférico com o intuito de mudar o estado da janela ou porta.

Palavras Chaves

Automação residencial, programação Java e C, microcontroladores, microcomputador.

SUMÁRIO

Capítulo 1 Introdução	39
Capítulo 2. APRESENTAÇÃO GERAL	41
Capítulo 3. DIAGRAMA DE FLUXO DE DADOS DO PROJETO.....	42
Capítulo 4. COMUNICAÇÃO	45
4.1. Topologia lógica da rede	45
4.2. Protocolo e controle de erros.....	46
4.3. Detecção de erros	47
4.4. Protocolo de acesso ao meio	48
4.5. Os quadros da rede	48
4.6. Os códigos dos quadros	51
4.7. Fluxo do protocolo	53
Capítulo 5. UNIDADE DE CONTROLE PERIFÉRICA	60
5.1. Introdução ao 8051	60
5.2. Principais vantagens do microcontrolador 8051	61
5.3. Fabricantes de microcontroladores.....	62
5.4. Linguagens de programação para o microcontrolador 8051	62
5.5. A unidade de controle utilizada.....	62
Capítulo 6 SOFTWARE DO MICROCONTROLADOR.....	64
Capítulo 7. SOFTWARE DO MICROCOMPUTADOR.....	68
7.1. Visão geral.....	68
7.2. Pré-requisitos do sistema	69
7.3. Os programas do servidor	69
7.4. Diagrama do modelo físico do banco de dados.....	70
7.5. Estudo de caso	71
7.5. Estudo de caso	71
7.6 Tela do programa do usuário.....	81
Capítulo 9. PRÉ-REQUISITOS DO SISTEMA.....	88
Capítulo 10. CONCLUSÃO	89
Bibliografia.....	91

Índice de Figuras

FIGURA 3.1 – DFD macro do sistema	12
FIGURA 3.2 – Estrutura do microcontrolador	13
FIGURA 4.1 – Topologia lógica da rede.....	16
FIGURA 4.2 - Fluxo principal – parte 1	24
FIGURA 4.3 – Fluxo principal – parte 2	24
FIGURA 4.4 - Recebendo dado da rede	25
FIGURA 4.5 – Interpretando quadro de Dado	26
FIGURA 4.6 – Interpretação do quadro de NAR.....	27
FIGURA 4.7 – Interpretação do quadro de TOKEN	28
FIGURA 4.8 – enviando dado na rede	29
FIGURA 5.1 - Diagrama de blocos do 8051	31
FIGURA 5.2 – Diagrama de blocos 80C552.....	33
FIGURA 6.1 - Diagrama de seqüência do programa implementado.....	36
FIGURA 7.1 – Modelo do banco de dados	40
FIGURA 7.2 – Pagina de envio de ordem ao sistema.....	51
FIGURA 7.3 – Inserindo novo hardware.....	51
FIGURA 7.4 Inserindo ordens ao periférico.....	52
FIGURA 7.5 Verificando estado do periférico.....	52
FIGURA 7.6 Pesquisa de hardwares instalados.....	53
FIGURA 7.7 Pesquisa de ordem do hardware.....	53
FIGURA 8.1 – Memória do sistema	54
FIGURA 8.2 - Controle que liga e desliga o motor e sensores de controle	55
FIGURA 8.3 – Controle do Sistema	55
FIGURA 8.4 – Circuito Completo	56
FIGURA 8.5 – Diagrama que une o periférico com o 8051.....	57

Índice de siglas

CI - Circuito Integrado

DFD - Diagrama de fluxo de dados

UCC - Unidade central de controle

UCP - Unidade central periférica

MIDP – *Mobile Information Device Profile*

LAN - *Local Area Network*

Capítulo 1. INTRODUÇÃO

De maneira geral, as residências classificadas como de padrão médio ou de alto padrão apresenta elevado número de portas e janelas, fato que traz grandes dificuldades para o deficiente com dificuldades de locomoção. Esse problema enfrentado pelos deficientes ainda não está sendo adequadamente considerado por projetistas, arquitetos, engenheiros ou construtoras que, atendem fundamentalmente às necessidades do usuário comum.

Constata-se essa tendência mercadológica principalmente em função da escassez de oferta de itens de conforto destinados aos deficientes com dificuldades de locomoção. Quando muito, o mercado disponibiliza aos deficientes paraplégicos rampas e elevadores em substituição às escadas e banheiros de uso comum adaptado.

O principal objetivo deste projeto é disponibilizar uma solução que permitirá ao usuário, sem necessidade de se locomover ou pedir para um terceiro, ter conhecimento do estado de sua residência, ou seja, verificar se as portas e as janelas estão abertas ou fechadas e executar o ato de abrir ou fechá-las.

A principal razão pela qual se desenvolveu este sistema é suprir parte da carência de oferta do mercado para os deficientes paraplégicos, visto que a automação irá atenuar o desconforto dos deficientes na medida em que se elimina a dependência de terceiros e o incômodo de locomoção para se executar uma simples tarefa como a de verificar e modificar o estado das portas ou janelas.

Embora o sistema tenha sido desenvolvido para atender a esse grupo da sociedade, nada impede que pessoas sem deficiência o utilizem, principalmente porque qualidades como conforto, comodidade e praticidade advindas do processo de automação atendem às necessidades não apenas de usuários específicos, mas de toda a sociedade.

Para que esse projeto pudesse ser desenvolvido, há vários itens/fatores relevantes que foram considerados, tais como, a evolução dos circuitos integrados, a evolução da microeletrônica e a utilização do *browser*, conforme detalhado a seguir.

Com a evolução dos CIs (circuito integrados), tornou-se possível desenvolver vários tipos de aplicações no mundo da microeletrônica. Entre tais

criações possíveis, está o desenvolvimento de equipamentos que auxiliam nas tarefas diárias do lar.

Graças à evolução da microeletrônica, é possível utilizar microprocessadores em diversos tipos de equipamentos. Os avanços da tecnologia possibilitam o uso cada vez mais intenso dos microcontroladores como PIC e 8051 que são utilizados em diversos equipamentos do dia-a-dia. Deve-se ressaltar que tais aplicações não requerem grandes recursos de memória nem grande velocidade de processamento e seu baixo custo representa fator significativo para sua utilização.

A utilização de *browser* é muito comum e conhecido principalmente em função da propagação do uso da internet. Esse projeto utiliza o mesmo programa da Internet para interagir o hardware com o usuário. O usuário vê o estado de cada periférico instalado e envia ordem via *browser*. A escolha dessa via foi feita para que o usuário pudesse ter a interface, a mais amigável possível.

Enfim, utilizando os recursos citados, este projeto simula a automação de uma parte uma casa. Utilizando o microcomputador e o microcontrolador, o usuário pode enviar uma ordem ou saber o estado de um periférico (agente executor das ordens do usuário) pelo programa do usuário. Além disso, quando esses elementos receberem uma determinada ordem, esta será executada e, assim, o deficiente poderá ter o controle de sua residência.

Capítulo 2. APRESENTAÇÃO GERAL

O projeto está dividido em dez capítulos. O primeiro capítulo é a introdução do projeto. O segundo capítulo mostra uma abordagem de como o trabalho será feito. O terceiro capítulo abordar a apresentação dos Diagrama de Fluxo de Dado (DFD) com o intuito de mostrar o macro-sistema desse projeto. Além disso, nesse tópico, será visto cada elemento DFD de forma superficial e, ao final, o leitor terá uma idéia do funcionamento do sistema.

O quarto capítulo descreverá a comunicação. Será especificado o protocolo utilizado, tamanho de quadros de dados, tipos de transmissão, entre outros detalhes.

No quinto capítulo, é abordada a unidade de controle. Nessa parte, são descritos os microprocessadores e suas funções no sistema.

No sexto capítulo, é abordado o funcionamento do *software* que controla os microprocessadores obedecendo aos pré-requisitos descritos. Para a implementação e devida demonstração do assunto, são mostrados outros diagramas de como o programa trabalha no sistema. Além desse *software*, é construído mais um, com a finalidade de interagir com o morador.

O sétimo capítulo descreve o *software* do microcomputador. Seus pré-requisitos, estudo de caso, diagrama do banco de dados são os dados descritos nesse capítulo.

O nono capítulo descreve os pré-requisitos do sistema como um todo. E por fim a conclusão do trabalho.

Capítulo 3. DIAGRAMA DE FLUXO DE DADOS DO PROJETO

Esse capítulo descreve, de forma geral, como o projeto está distribuído. São vistos vários diagramas de fluxo de dados (DFD) que mostram os componentes como unidades de controles, periféricos e o microcomputador do morador. Cada elemento é descrito de modo a sanar as dificuldades de interpretação do sistema. Assim, a próxima etapa compreende a visualização dos DFD, reproduzido a seguir.

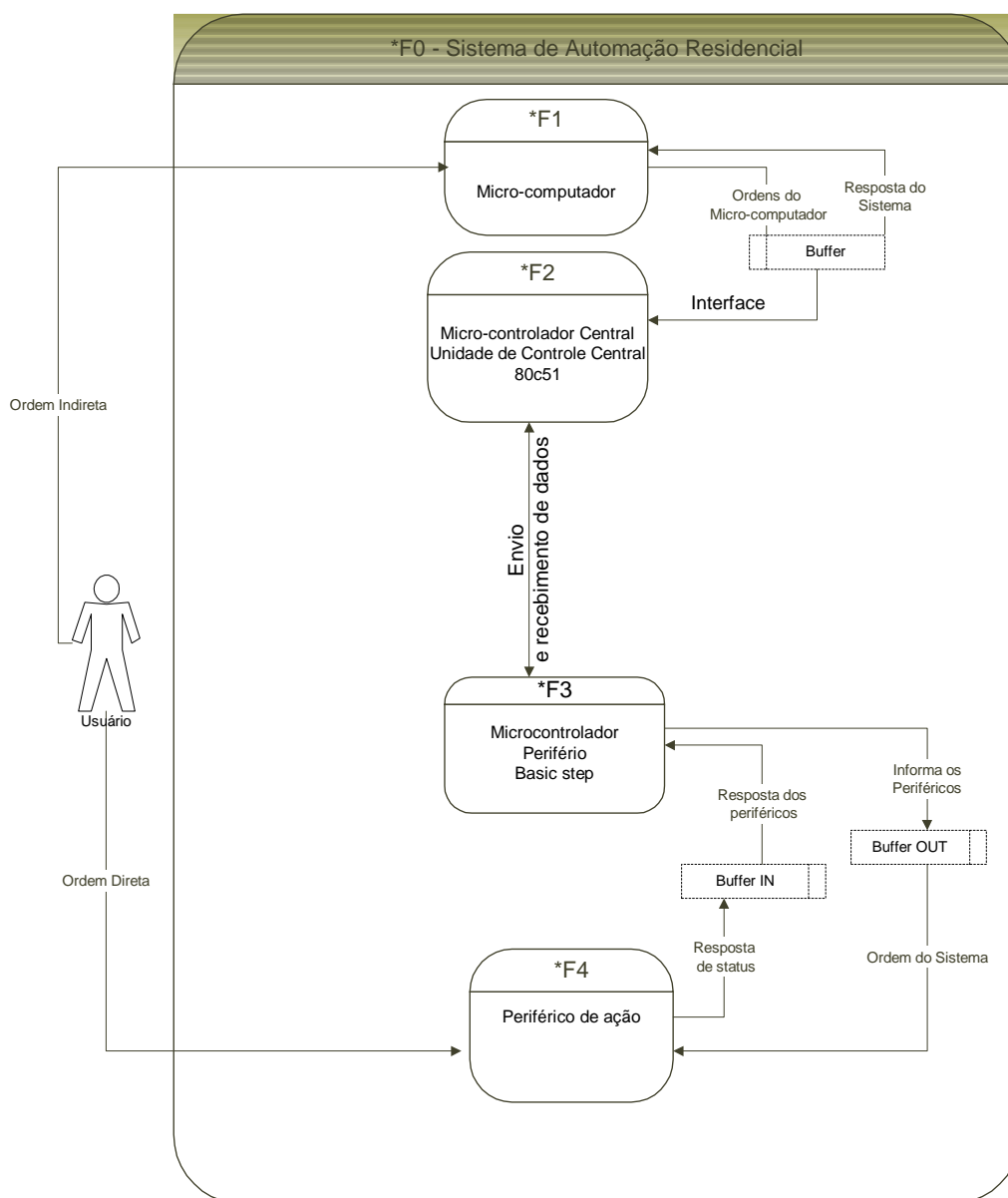


FIGURA 3.1 - DFD macro do sistema

Na figura 3.1, há quatro blocos:

1. Microcomputador
2. Unidade de Controle Central (UCC)
3. Unidade de Controle Periférica (UCP)
4. Periférico

O microcomputador, denominado de servidor, é responsável pela inteligência do sistema. Será composto de servidor WEB, banco de dados e *softwares* que controlam a interface com o usuário e o estado dos periféricos. Esse último é controlado por meio um de *software* que tem a responsabilidade de alterar o estado do periférico no banco de dados, enviar ordens e verificar estado de periféricos. O *software* que interage com usuário tem sua visualização por meio de navegadores de Internet. O acesso a este *software* pode ser feito por meio de uma rede interna ou no próprio servidor. O usuário tem a possibilidade de executar suas ordens por meio do microcomputador ou no próprio periférico como ele desejar.

Enviar ordem, verificar estado, verificar periférico e verificar as ordens instaladas nos periféricos são as funcionalidades desse *software*. O usuário poderá ter acesso às informações no servidor ou em qualquer outro microcomputador ligado a uma rede. Abaixo o diagrama que demonstra a estrutura explicada.

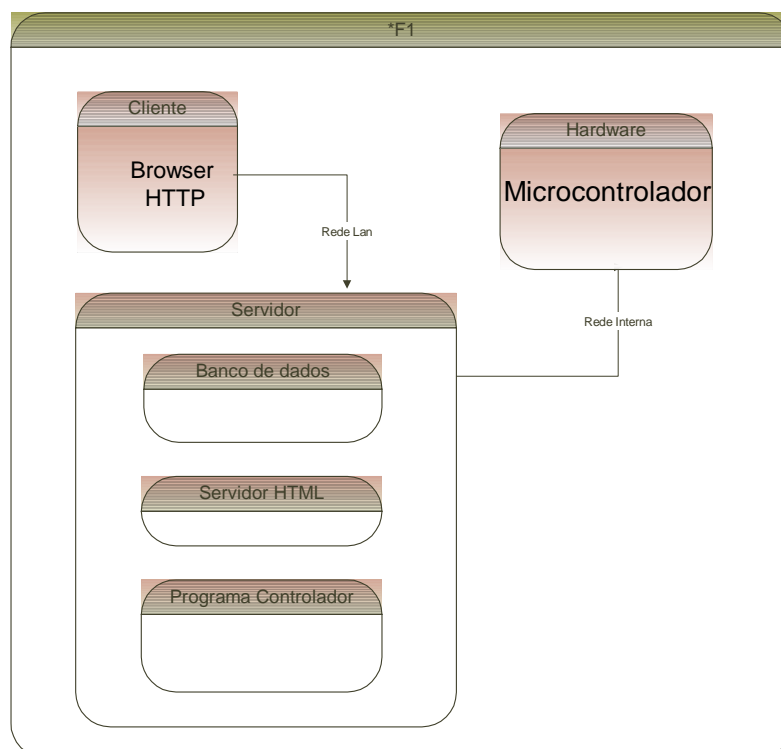


FIGURA 3.2 - Estrutura do microcomputador

A UCC e UCP são responsáveis pelo envio do dado ao periférico. A UCC terá, na sua memória, os estados atualizados de todos os periféricos instalados. A UCP processará as ordens recebidas pela UCC, enviará uma resposta quando solicitada e enviará ordens para os periféricos. A solicitação irá ocorrer com base no fluxo do protocolo que será descrito no capítulo três.

O periférico terá a função de executar os dados vindos do usuário e retornar as devidas respostas quando solicitadas. O usuário poderá acionar o periférico localmente e, conseqüentemente, alterar o estado do periférico.

Para demonstração, foi implementada, nesse projeto, uma solução simples de um motor que gira no sentido horário ou anti-horário com base na ordem solicitada pelo usuário. Uma UCP e um periférico simples são implementados com o intuito de demonstrar que a simulação atingiu seu êxito. A UCC não será implementada, neste momento, por se tratar de objeto de pesquisa para próximos projetos.

Deve-se ressaltar que a rede LAN que foi citado nesse capítulo não se faz necessária para o correto funcionamento, basta um micro que funcionará com servidor. A utilização da rede LAN só se faz necessária caso o morador deseje maior conforto em sua residência.

Capítulo 4. COMUNICAÇÃO

Este capítulo descreve o protocolo utilizado na rede de automação residencial de minha autoria. Os quadros que trafegam na rede, o fluxo do protocolo bem como as interpretações de cada campo serão descritas nesse capítulo.

4.1. Topologia lógica da rede

A topologia lógica da rede deste trabalho é do tipo árvore de nível quatro interligado com a topologia anel hierárquica. O servidor de aplicações está no nível zero da árvore e pertence ao anel denominado de número um. O servidor envia um quadro para cada UCC – Unidade de Controle Central - obedecendo a uma tabela de pendências estipulada pelo usuário para sistema automação residencial. O servidor somente tem acesso direto as UCC. Entende-se como UCC a unidade que está entre o servidor e o resto do sistema. Essa unidade armazena, na sua memória, o estado atualizado dos periféricos que ela pode controlar, pois, para atualizar o banco de dados, o servidor requisita as informações a esse componente. Além disso, a unidade tem a responsabilidade de enviar as requisições do servidor para a respectiva UCP (Unidade Central Periférica) que será descrito a seguir. Convém esclarecer que a UCC não é o enfoque desse trabalho.

O segundo anel tem a UCP - unidade controladora do periférico - e a UCC. O terceiro anel é constituído da unidade controladora do periférico que tem como host os próprios periféricos. A UCP tem a responsabilidade de enviar e receber as requisições da UCC para o periférico solicitado.

A hierarquia da rede é do tipo hierárquico-distribuído, pois cada elemento pertence à hierarquia. Na figura três, cada hierarquia é representada por anéis, sendo ao todo três anéis, nesse caso. A distribuição existe, pois o controle de cada elemento está segmentado de acordo com o anel. Por exemplo, no primeiro anel o servidor é o servidor das informações e não deve se comunicar diretamente com a UCP ou Periférico. Para obter essas informações, é preciso questionar a UCC que retorna as informações do respectivo UCP. Para obter as informações do periférico,

os mesmos passos devem ser seguidos. O servidor deve questionar a UCC que, então, questiona a UCP até chegar ao periférico, de forma hierárquica. O retorno da informação obedece ao mesmo caminho inverso.

Essa rede não tem qualquer redundância de tal forma que se uma malha da rede para de funcionar não há outro via de segurança, pois o pré-requisito desse sistema é ter uma malha de tal forma que o anel não deixe de funcionar.

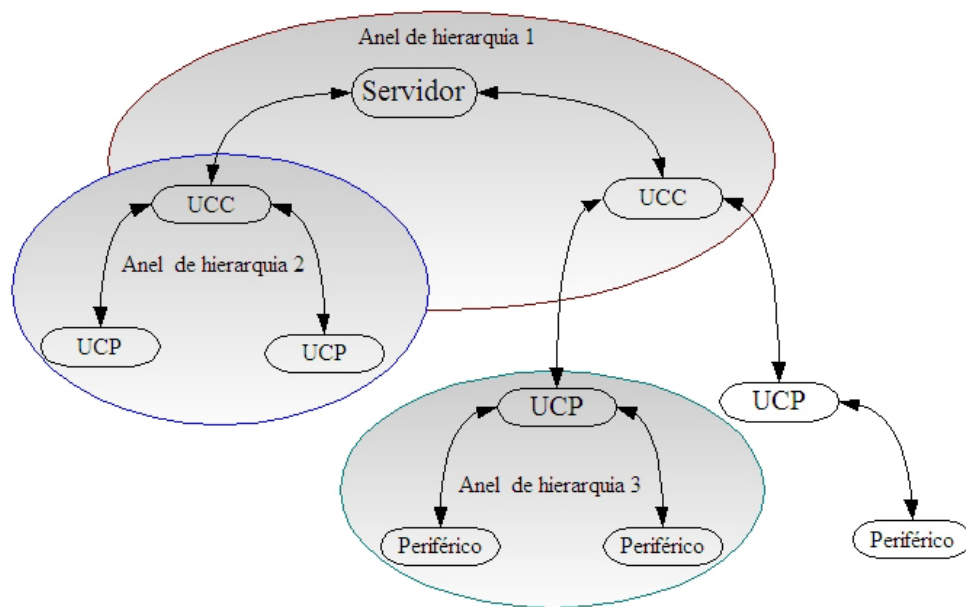


FIGURA 4.1 - Topologia lógica da rede

4.2. Protocolo e controle de erros

O protocolo de controle de acesso ao meio utilizado é ordenado sem contenção. Entende-se como sem contenção, os protocolos de acesso ordenando ao meio de comunicação, de modo a evitar o problema da colisão.

Os dados são transmitidos via Broadcast. Por essa razão, todos os componentes que estiverem compartilhando a mesma linha de comunicação recebem os dados. As estações conectadas à rede somente transmitem quando interrogadas pelo controlador da rede, que é uma estação centralizadora. Se não tiver quadro para transmitir, o nó interrogado envia um quadro de estado, simplesmente avisando ao controlador que está em operação [8].

Para garantir a confiabilidade dos quadros enviados, foram criadas algumas variáveis: o NAR (Negative acknowledgement with retransmission), que tem a finalidade de avisar ao requisitante que o quadro enviado apresentou algum erro; a memória com o número de próximo quadro em que cada quadro enviado recebe uma numeração que deve ser validada; por fim, foi criada a paridade que tem a finalidade de verificar a consistência do quadro. Esses são os dispositivos de controle de erros que serão explicados com mais detalhes a seguir [8].

O N.A.R. funciona como um semáforo para o transmissor. Caso o quadro recebido pelo receptor esteja errado, o receptor enviará um quadro com a não confirmação da mensagem. Assim, o servidor, ao recebê-lo, terá que enviar o mesmo quadro de volta.

Se, porventura, ocorrer alguma falta de sincronismo e vários quadros de NAR forem enviados ao transmissor, a comunicação com aquele receptor será encerrada. Deve-se ressaltar que a implementação atual não há nenhum tipo de redundância caso algum link dessa rede pare de funcionar [6].

4.3. Detecção de erros

A camada de enlace será a responsável pela detecção do erro e solicitação de uma retransmissão do mesmo quadro. No caso específico, a camada está preocupada com a integridade do quadro.

A detecção do erro é feita por meio da paridade par dos bits de cada quadro.

A paridade tem a função de verificar se o quadro transmitido é igual ao recebido pelo cliente. Essa forma de detecção de erro adiciona um bit no quadro. O valor do bit é escolhido de forma a deixar uma quantidade par (paridade par) de bits ou ímpar (paridade ímpar). Nesse projeto, foi implementada a paridade par [8].

Para exemplificar o algoritmo de paridade, imagine um quadro contendo o valor “100101”. Utilizando a paridade par, é acrescentado mais um bit com valor 1 no final do quadro. Portanto, o novo quadro é “1001011”. O receptor deverá contar uma quantidade par de uns. Caso a quantidade não seja a esperada, o quadro terá erro e deverá ser descartado. No caso desse protocolo, um quadro de NAR deve ser enviado para o requisitante.

4.4. Protocolo de acesso ao meio

O protocolo de acesso ao meio é com passagem de permissão (TOKEN) que funciona como permissão de transmissão de dados. A rede funciona do endereço menor para o maior da interface de rede. A cada passagem de permissão – TOKEN - o equipamento que desejar transmitir tem que alterar os bits do quadro de TOKEN e isso simboliza que o devido periférico deseja transmitir algum quadro. E assim, perpetua-se essa lógica até que chegue na interface de número menor e a tabela é reiniciada novamente. Esse é o ciclo do TOKEN [8].

4.5. Os quadros da rede

Abaixo estão descritos os quadros que estarão em uso na transmissão de dados. Os tamanhos dos quadros e a quantidade de bits que cada parte do quadro utilizam estão descritos abaixo.

PC -UCC (modo de transmissão ASCII)

Dado:

Esse é o utilizado entre o servidor e a UCC quando o protocolo deseja transmitir algum dado.

Endereço	Número do quadro	Flag Dado	Endereço U. CP.	Nº do periférico	Tabela de estado	Vazio	Paridade

N.A.R.

Se algo de errado ocorre, esse é o quadro de NAR que tem a finalidade de enviar ao requisitante aviso de que ocorreu algum erro na transmissão.

Endereço	Nº do quadro	Flag N.A.R	estado	vazio	paridade

TOKEN:

Esse quadro tem a finalidade de perguntar ao destinatário se há algo a transmitir.

Endereço	Nº do quadro	<i>Flag TOKEN</i>	Estado	Vazio	Paridade

Controle:

Esse quadro tem a finalidade de abrir e fechar a comunicação com o destinatário.

Endereço	Nº do quadro	<i>Flag Controle</i>	Estado	vazio	Paridade

UCP - Periférico

Saída da UCP

Transmissor UCP: ordem

Esse é o utilizado entre o periférico e a UCP quando o protocolo deseja transmitir algum dado.

Endereço	Nº do quadro	<i>Flag Dado</i>	Dado (ordem)	Paridade
5 bits	2 bits	2 bits	6 bits	1 bit

Total de bits : 16bit

TOKEN

Esse quadro tem a finalidade de perguntar ao destinatário se há algo a transmitir.

Endereço	Nº do quadro	<i>Flag TOKEN</i>	estado	vazio	Paridade
5 bits	2 bits	2 bits	3 bits	3 bits	1 bit

Total de bits : 16 bits

UCC – UCP

Dado:

Esse é o utilizado entre o servidor e a UCC quando o protocolo deseja transmitir algum dado.

Endereço	Nº do quadro	Flag dado	Número do periférico	Dado (ordem)	Vazio	Paridade
5 bits	2 bits	2 bits	5 bits	6 bits	3 bits	1 bits

Total de bits : 24bits

N.A.R.

Se algo de errado ocorre, esse é o quadro de NAR que tem a finalidade de enviar ao requisitante o aviso de que ocorreu algum erro na transmissão.

Endereço	Número do quadro	Flag N.A.R.	Estado	Vazio	paridade
5 bits	2 bits	2 bits	3 bits	3 bits	1 bits

Total de bits : 16 bits

TOKEN:

Esse quadro tem a finalidade de perguntar ao destinatário se há algo a transmitir.

Endereço	Número do quadro	Flag TOKEN	Estado	vazio	Paridade
5 bits	2 bits	2 bits	3 bits	3 bits	1 bits

Total de bits : 16 bits

Controle:

Esse quadro tem a finalidade de abrir e fechar a comunicação com o destinatário.

Endereço	Número do quadro	Flag Controle	Eestado	vazio	Paridade e
5 bits	2 bits	2 bits	3 bits	3 bits	1 bits

Total de bits : 16 bits

4.6. Os códigos dos quadros

Interpretação do código dos quadros já descrito.

1. Endereço

- Cinco bits com a numeração binária para identificar cada periféricos.
- Os bits são comparados por meio de circuito lógico chamado de ou exclusivo.
- Quando todos bits do quadro de endereço forem, um deve ser aceito por qualquer periférico. Assim, esse é o BROADCAST da rede.
- Se os bits dessa parte do quadro forem zero, nada ocorre.
- Se o endereço do periférico for igual a zero, isso significa que a tabela de estado deve ser toda enviada para o requisitante do pedido.

2. Estado

- **TOKEN**
 - Bits 011
 - Isso significa que o quadro pede verificação da tabela de estado.
 - Bits 001
 - Há mudanças na tabela de estado; o retorno para o solicitador do quadro é estado igual 001.
 - Bits 010
 - Há mudanças na tabela de estado; o retorno para o solicitador do quadro é 010.
- **Controle**
 - Se o número do quadro for zero, o estado informa o número do quadro do destinatário e isso abre a comunicação.
 - Bits 000
 - Fecha a conexão com o destinatário
- **NAR**
 - Bits 001

- Problema no pacote
- Bits 010
 - Erro no número do quadro
- Bits 011
 - Sem utilidade
- Bits 100
 - Sem utilidade

3. Flag

- Flag de N.A.R. (Negative acknowledgement with retransmission) - os dois bits têm o código igual a 10.
- Flag TOKEN - os dois bits são iguais a 01.
- Flag de controle - os dois bits são iguais a 11.
- Flag de dado - os dois bits serão iguais a 00.

4. Paridade

- O bit é zero ou um, de acordo com a somatória dos bits do quadro.

5. Número do quadro

- Os dados obedecem ao código binário, pois assim o demultiplexador interpreta a informação para a unidade de controle responsável executar a ordem.
- Se todos bits forem zeros, nenhum dado é processado, porque esse número é o DEFAULT.
- Se todos os bits forem iguais a um, isso significa o fim da tabela e a conexão pode ser fechada.

6. Dados

- Os dados obedecem ao código binário, pois, assim, o demultiplexador interpreta a informação para a unidade de controle responsável executar a ordem.
- Se todos bits forem zeros, nenhum dado é processado, porque esse número é o DEFAULT.

- Se todos os bits forem iguais a um, isso significa o fim da tabela e a conexão pode ser fechada.

7. Tabela de Estado

- Contém a ordem em que o periférico se encontra no sistema.

4.7. Fluxo do protocolo

O fluxo do protocolo que está sendo executado no projeto está descrito a seguir.

As figuras 4.2 e a 4.3 apresentam as principais rotinas de processamento, pois demonstram como o autômato do protocolo funciona. O protocolo inicia-se verificando o estado de cada periférico, de modo a atualizar seu estado no banco de dados. Essa rotina pode ser vista na figura quatro. Em seguida, o sistema verifica se há alguma pendência no sistema. Caso haja, o sistema abre a conexão com o periférico e envia a pendência, deve-se entender como pendências as ordens possíveis de serem enviadas pelo usuário. No outro caso, o protocolo só verifica o estado atual do periférico. A figura 4.3 mostra como esse fluxo funciona. Deve-se entender pendências as ordens enviadas pelo usuário.

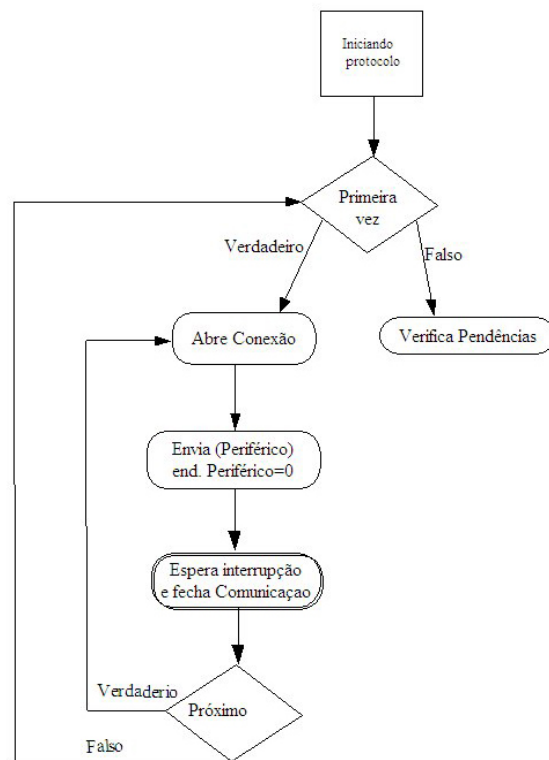


FIGURA 4.2 - Fluxo principal – parte 1

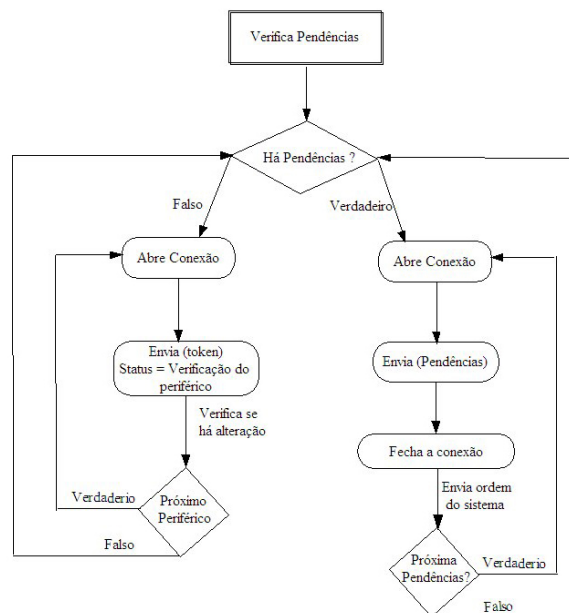


FIGURA 4.3 - Fluxo principal – parte 2

A figura 4.4 demonstra como o quadro é interpretado pelo sistema e envia as informações para as rotinas de TOKEN, NAR, DADO e ANALISA ESTADO. Essa figura demonstra o local onde ocorre a validação da paridade, verificação do endereço e número do quadro.

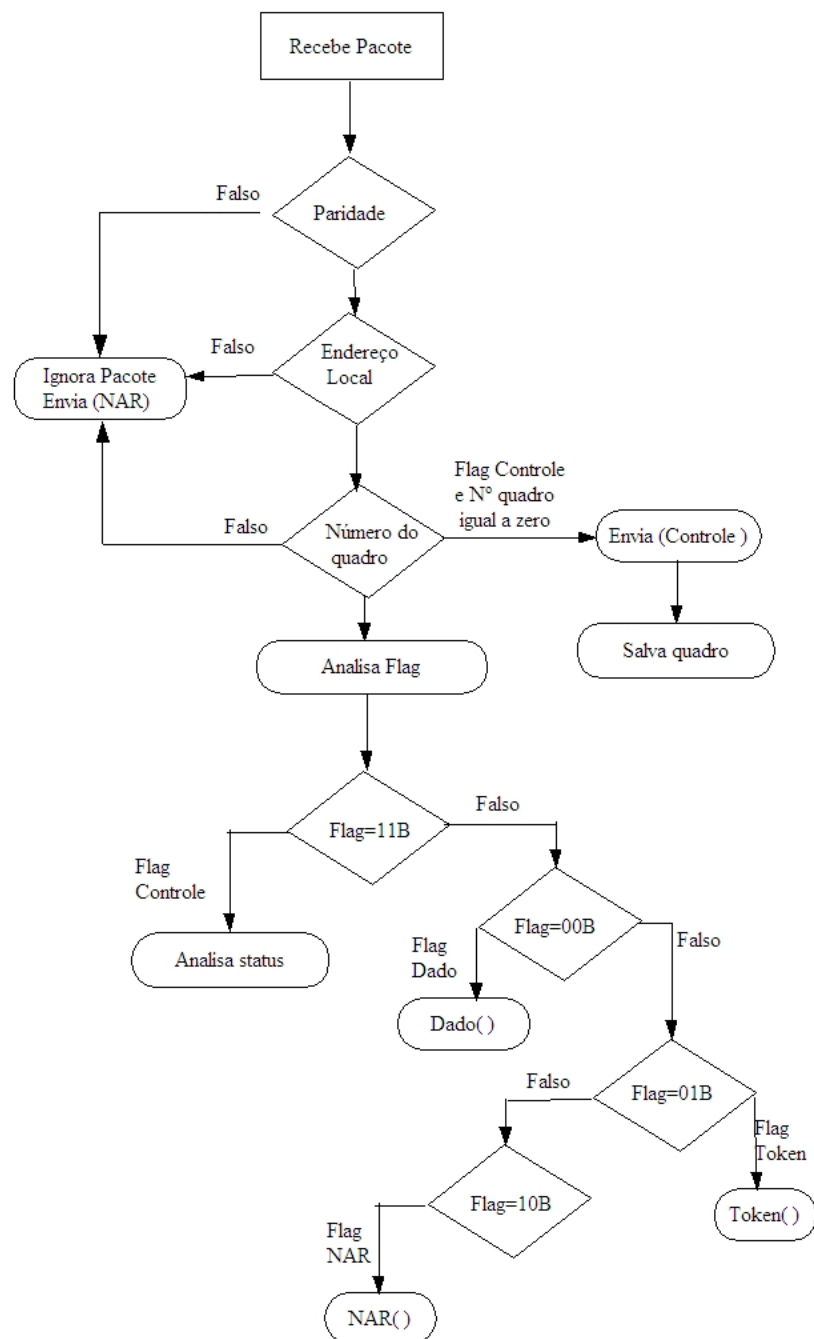


FIGURA 4.4 - Recebendo dado da rede

A figura 4.5 mostra como é processado o quadro de dados pelo protocolo. Nessa figura, é possível visualizar onde o endereço do periférico é interpretado, onde as ordens são enviadas para alterar o estado do periférico e o local onde o protocolo solicita o envio da toda a tabela com o intuito de atualizar o banco de dados do sistema.

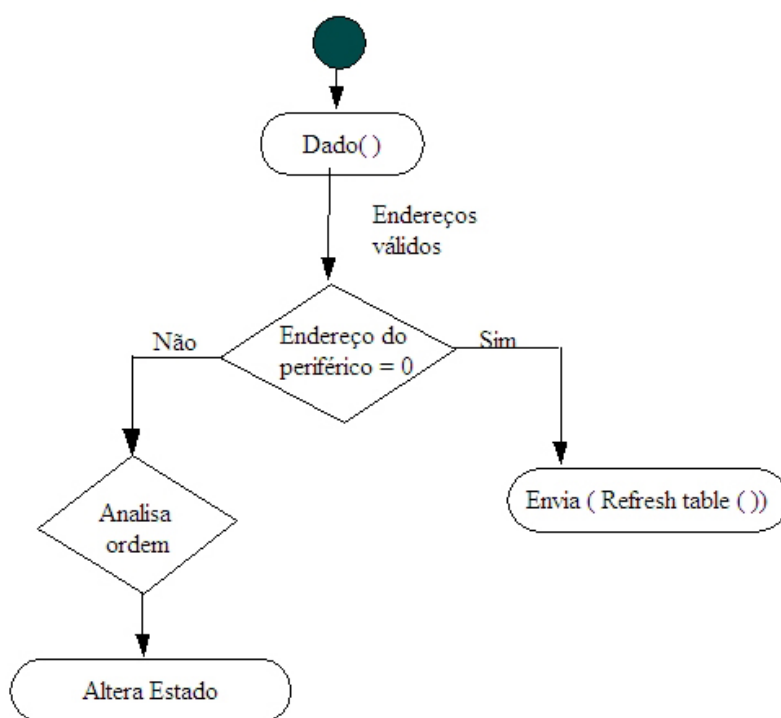


FIGURA 4.5 - Interpretando quadro de Dado

A figura 4.6 demonstra a rotina quando um quadro de NAR é recebido pelo protocolo. Nessa figura, vê-se a quantidade de vezes que o protocolo reenvia as informações e o que acontece quando a quantidade de reenvios supera a dez tentativas.

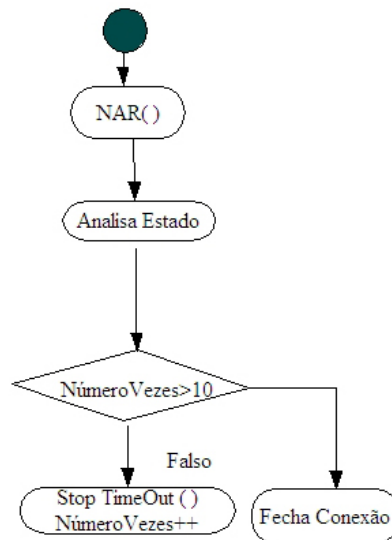


FIGURA 4.6 - Interpretação do quadro de NAR

A figura 4.7 demonstra como o TOKEN é processado e o que pode acontecer quando o estado é analisado. Vêm-se três estados possíveis: o primeiro 01 confirma que o destinatário tem algo para enviar para o receptor que deve se preparar para recebê-lo; o segundo estado é o 10 que diz que o destinatário não tem nada para enviar; por último, o estado 11 que solicita ao protocolo verificar sua tabela de estado. Caso haja a necessidade de envio de dados, um TOKEN de estado 01 é enviado e o dado é transmitido em seguida. Se não há nada para ser transmitido ao servidor, o protocolo só envia o TOKEN com o estado de 10. A figura de número doze mostra o fluxo de TOKEN completo.

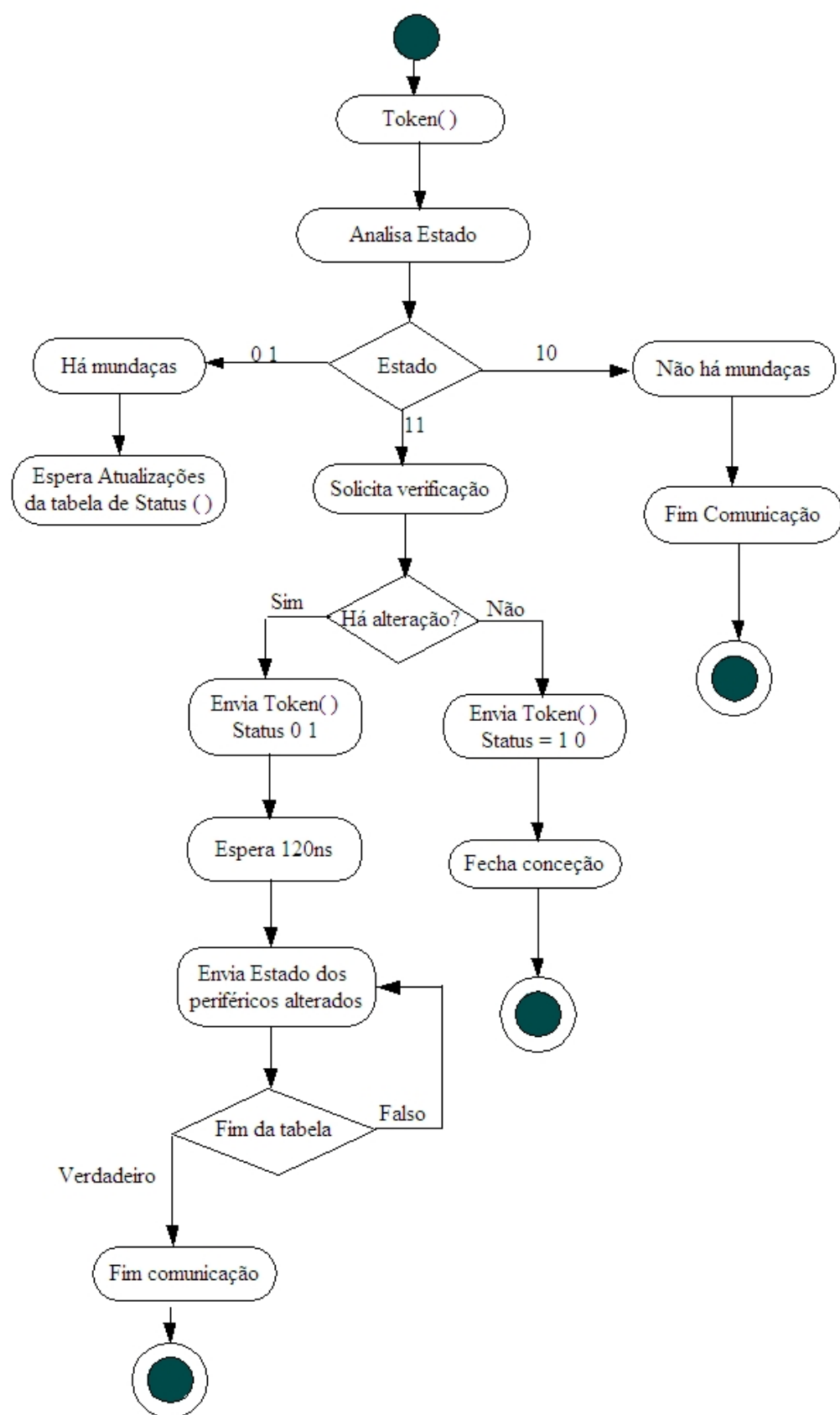


FIGURA 4.7 - Interpretação do quadro de TOKEN

A figura 4.8 demonstra os envios dos dados, a quantidade reenvios possíveis, a espera por uma NAR e o Exception() que ocorre quando a quantidade de reenvios é maior que três tentativas.

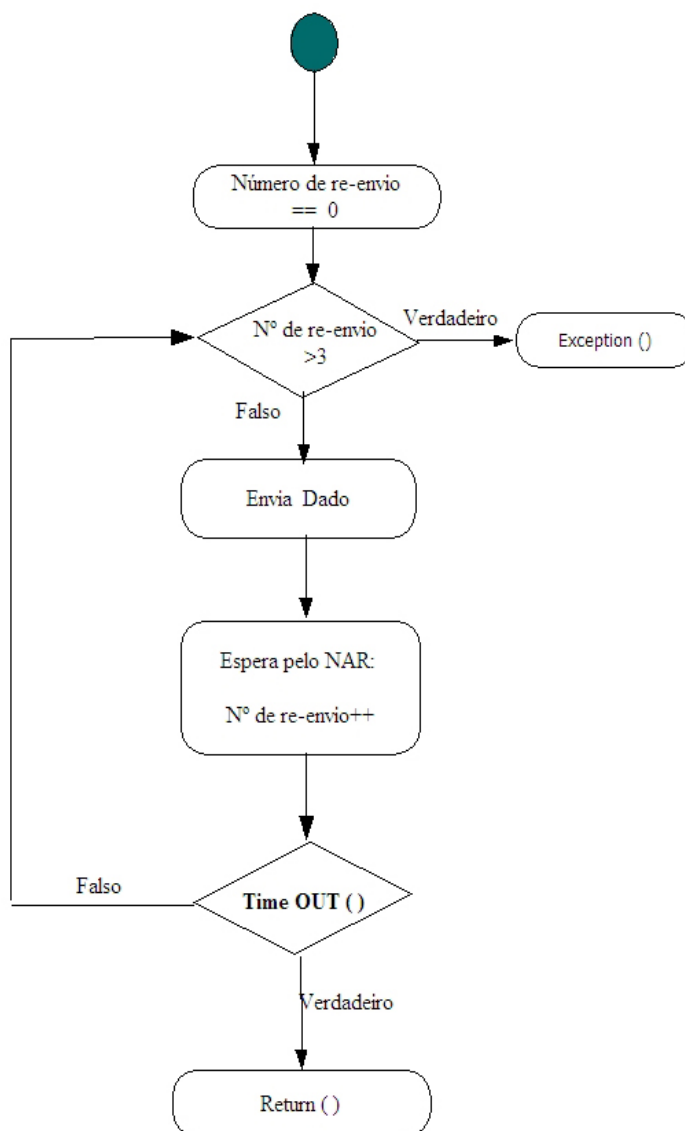


FIGURA 4.8 - Enviando dado na rede

Capítulo 5. UNIDADE DE CONTROLE PERIFÉRICA

5.1. Introdução ao 8051

O microcontrolador modelo 8051 é o membro da família MCS-51 que apresenta as seguintes características:

- CPU de 8bits otimizada para aplicações em controle.
- Alta capacidade de processamento booleano (manipulação de variáveis de um bit).
- 32 linhas de E/S bidirecionais e individualmente endereçáveis.
- 128 bytes de RAM interna destinada a dados.
- RAM de baixo consumo de energia.
- 2 temporizadores/contadores de 16bits.
- UART full duplex.
- 5 estruturas de interrupção com 2 níveis de prioridade .
- Clock interno.
- 4 Kbytes de memória de programa interna .
- 64 Kbytes de memória de programa endereçáveis.
- 64 Kbytes de memória de dados endereçáveis.
- Frequência de clock entre 1.2 MHz e 12 MHz.

A arquitetura básica do microcontrolador da família MCS-51 tem o seguinte diagrama de blocos: [3]

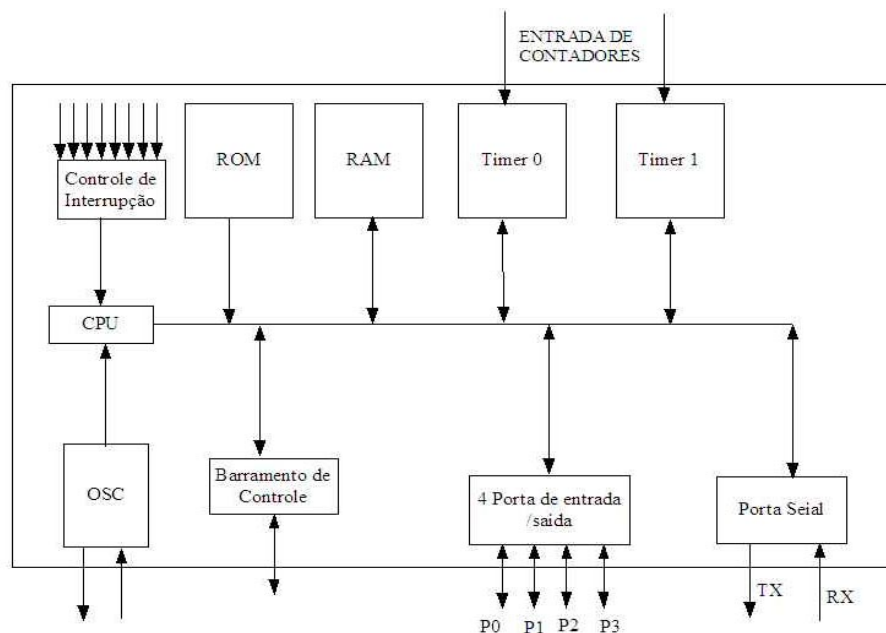


FIGURA 5.1 - Diagrama de blocos do 8051

5.2. Principais vantagens do microcontrolador 8051

O microcontrolador dessa família foi escolhido pelas seguintes razões:

- **Popular:** uma gama completa de produtos de suporte está disponível no mercado sem custos (free), proporcionando economia real em termos de ferramentas de treinamento e suporte para os programas.
- **Rápido e eficaz:** a arquitetura se correlaciona de perto com o problema que está sendo solucionado. As instruções especializadas significam que menos bytes precisam ser buscados e menos jumps condicionais são processados.
- **Baixo custo:** alto nível de integração do sistema em um único componente e a pouca necessidade de componentes para criar um sistema que funcione.

- **Compatibilidade:** os códigos binários são os mesmos para todas as variações fabricadas do 8051, diferente de outras famílias de microcontroladores.
- **Multi Sourced:** mais de 12 fabricantes.
- **Aperfeiçoamentos constantes:** melhorias na fabricação aumentam a velocidade e reduzem o consumo de potência. Há ainda versões de 16 bits, de diversos fabricantes.

5.3. Fabricantes de microcontroladores

A seguir, relação de fabricantes que produzem modelos similares ao MCS-51: Motorola, Mitsubishi, Siemens, NEC, Hitach, Philips, Intel, Microchip, Matsushitta, Toshiba, National, Semiconductor Zilog, Texas Instruments, Sharp.

5.4. Linguagens de programação para o microcontrolador 8051

As linguagens mais utilizadas na programação da família MCS-51 são C, BASIC e Assembly. A maior parte da literatura sobre programação deste microcontrolador utiliza a linguagem Assembly.

5.5. A unidade de controle utilizada

A unidade de controle periférica é um microcontrolador modelo 80C552 da PHILIPS. A comunicação serial e a porta P4 são os meios de entrada e saída de dados desse periférico.

O microcontrolador 80C552 é fabricado com a tecnologia CMOS de processo. A base desse microcontrolador é a família 8051 descrita nos tópicos 4.1 a 4.4. Assim, o conjunto de instruções do microcontrolador 80C552 é o mesmo da família 8051 [14].

O 80C552 tem uma memória ROM de programa de 8Kx8 e 256x8 de memória RAM. Maiores informações desse microcontrolador podem ser encontradas no *site* <http://www.semiconductors.philips.com/pip/P80C552.html>, da PHILIPS LOGIC. A figura a seguir mostra a arquitetura desse modelo.

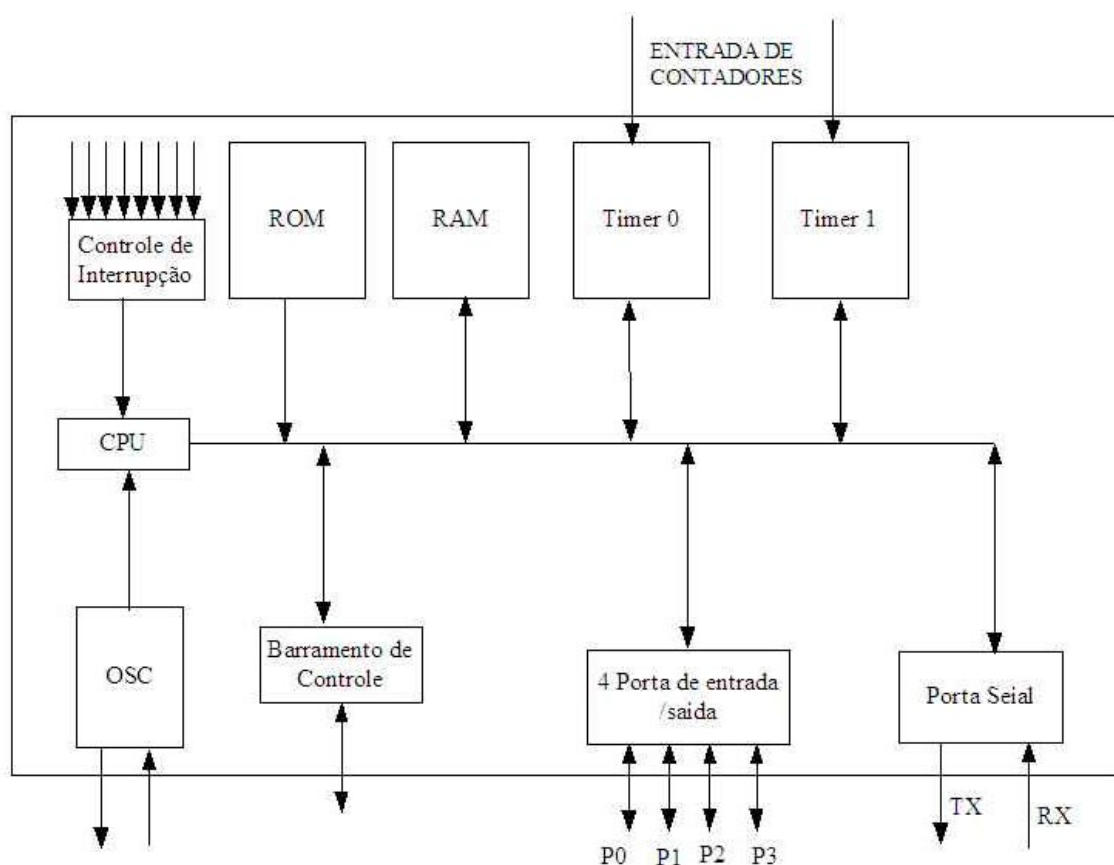


FIGURA 5.2 - Diagrama de blocos do 80C552

Capítulo 6. SOFTWARE DO MICROCONTROLADOR

O *software* do microcontrolador foi produzido na linguagem C de programação. O aplicativo que fez a compilação foi o SDCC, *Small Device Compiler* que pode ser encontrado no *site* <http://sdcc.sourceforge.net/>. SDCC é um *software* free que obedece ao padrão ANSI –C *compiler*. O SDCC dá suporte para as seguintes unidades centrais de processamento:

- no Intel 8051,
- Maxim 80Ds390
- Zilog Z80
- Motorola 68HC08
- Microchip PIC14 E PIC16

Nesse projeto, o programa foi compilado para funcionar na CPU Intel 8051. O programa implementado interpreta os dados vindos do servidor pela interrupção da porta serial do microcontrolador e envia para o periférico através da porta P4 do microcontrolador. Há também o caso do servidor requisitar informações. Nesse caso, o microcontrolador envia informações para o servidor como, por exemplo, o estado do periférico.

A porta P4, composta de oito bits, faz a interface do periférico. A porta P4 é endereçada bit a bit e seus nomes são: P4.0, P4.1, P4.2, P4.3, P4.4, P4.5, P4.6 e P4.7. Esse projeto utiliza os quatros primeiros bits. As portas P4.0 e P4.1 verificam o estado dos sensores e P4.2 e P4.3 enviam a ordem para o periférico.

Os sensores podem ter os seguintes valores binários 10_B (aberto), 01_B (fechado) e 00_B (não definido) em que cada valor representa um determinado estado. As ordens possuem os seguintes valores possíveis 10_B (abrindo), 01_B (fechando).

A seguir, está demonstrado o diagrama de seqüência da funcionalidade do sistema implementado. Esse diagrama demonstra os métodos que serão executados no caso da melhor hipótese do sistema. Deve-se ressaltar que a base da implementação desse sistema foi o protocolo descrito anteriormente.

A linha vertical é chamada de linha de vida que representa a vida do programa durante sua execução. Cada ação é representada por uma flecha entre as linhas verticais. Cada ação é rotulada com o nome do método e seus argumentos a ser executado no programa.

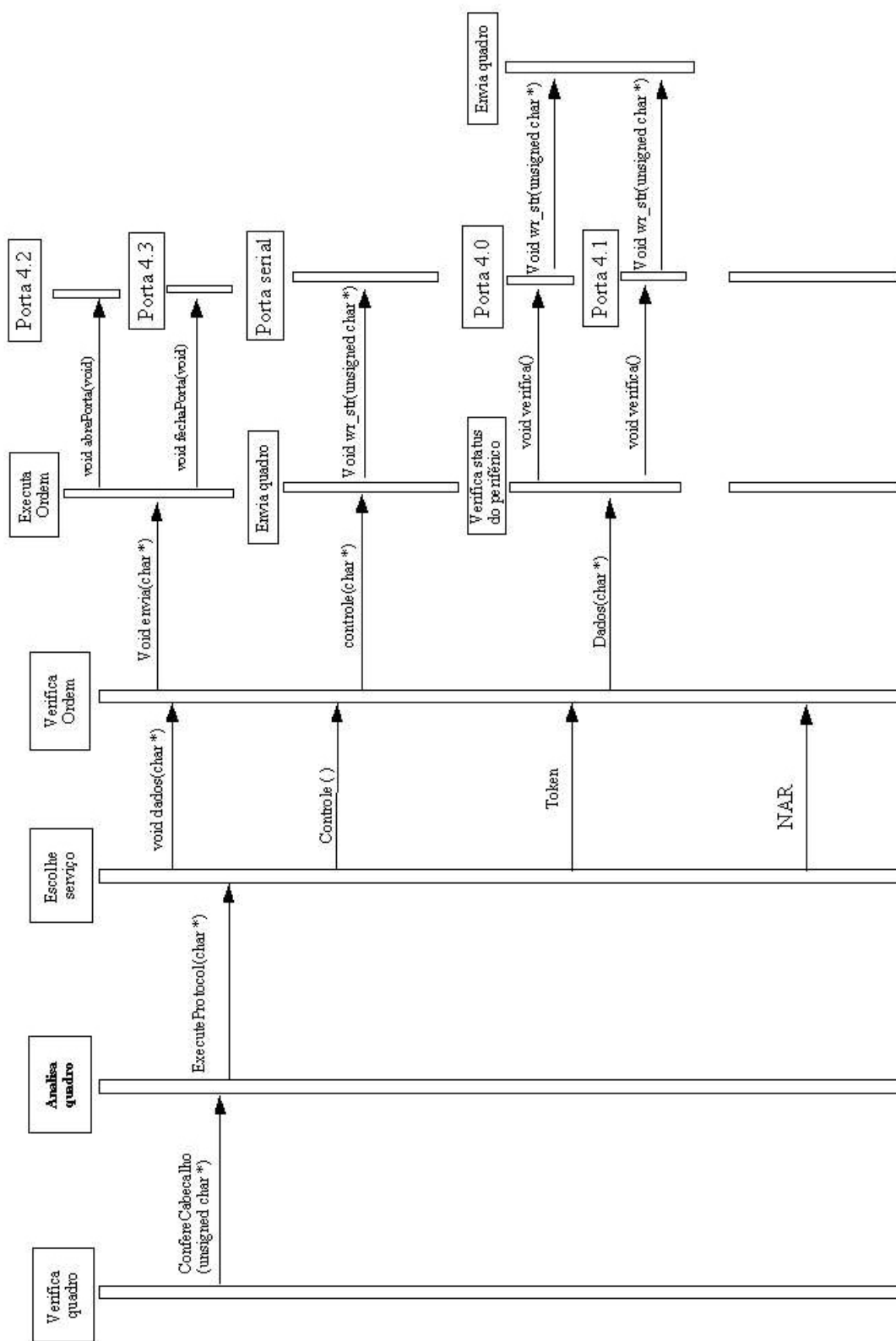


FIGURA 6.1 - Diagrama de seqüência do programa implementado.

O presente capítulo apresentou a descrição:

- de como o protocolo foi compilado;
- do *software* utilizado para este fim;
- das portas utilizadas - como as portas P4 e a porta serial;
- das funcionalidades de cada bit;
- do fluxo de métodos utilizados para a implementação e quando são executados.

Capítulo 7. SOFTWARE DO MICROCOMPUTADOR

7.1. Visão geral

Há dois *softwares* que funcionam concorrentemente e tem funcionalidades distintas. O primeiro tem a finalidade de interagir o usuário com o sistema de automação residencial. O outro é puramente de controle com a finalidade de gerenciar os periféricos.

O programa que gerencia a interface entre o usuário tem como base algumas tecnologias da internet. Vale a pena ressaltar que algumas características importantes como segurança estão aquém das reais necessidades de um *software* que tem esse fim, pois o objetivo do programa, nesse momento, é poder rodar em máquinas locais e ter uma possibilidade de, no futuro, colocá-la em rede.

A base desse *software* está na tecnologia Struts® oferecido pela The Apache Jakarta Project®. Essa tecnologia tem seu código aberto e não requisita nenhum pagamento ao detentor da mesma. O Struts® funciona com base na linguagem Java® de programação. Assim, os códigos fontes desse *software* de automação estão nessa linguagem. Para o correto funcionamento dessa tecnologia, é preciso fazer o *download* de um *software*, cujo é um servidor de web. Seu nome é o TomCat® que também é um *software* livre oferecido pela The Apache Jakarta Project®.

O *software* que gerencia os periféricos foi construído com base no pacote J2SDK (Java 2 Software Development Kit) produzido pela Sun® microSystems. Esse *software* é responsável pela implementação de boa parte do protocolo desse projeto final.

7.2. Pré-requisitos do sistema

Para o usuário poder visualizar e utilizar o programa é preciso algum aplicativo que leia páginas da Web como o Internet Explorer da Microsoft® ou Netscape®.

Há mais um software, o servidor MY SQL de banco de dados, que é pré-requisito, pois toda a configuração estará sendo aplicada dentro de tabelas. A sua documentação será a visualização do modelo físico do banco.

Logo abaixo está uma lista de softwares necessários.

- J2SDK® (Java 2 Software Development Kit)
- TomCat®
- MySQL®
- Pacote de Struts®
- Navegadores de Internet

7.3. Os programas do servidor

A comunicação física a ser utilizada com o microcontrolador é a mesma da porta serial dos microcomputadores, isto é, a modularização obedece ao padrão RS232 e os dados são enviados serialmente.

Para a implementação dessa comunicação, é utilizado o pacote Java COMM distribuído pela Sun® *microSystems* que possibilita essa comunicação.

O protocolo utilizado e os respectivos dados que transitam na linha de comunicação estão especificados no capítulo de comunicação no item protocolo.

Há um aplicativo que está funcionando no servidor TomCat®. Esse é encarregado de interagir com o usuário recebendo ordem e mostrando configurações e estado do sistema. O usuário poderá visualizar esse aplicativo em qualquer *browser* se o servidor estiver funcionando. Há outro software encarregado

de enviar e receber os dados da unidade centrais de processamento e executar as devidas mudanças no banco de dados.

Assim, esse software está dividido em dois blocos bem distintos: o primeiro é a comunicação com os periféricos; o segundo, a interação com o usuário.

7.4. Diagrama do modelo lógico do banco de dados

Nesse tópico, é mostrado o modelo lógico do banco de dados. Esse banco de dados contém os valores de todo o sistema. Descrição do periférico, tipo de ordens e pendências são exemplos de dados armazenados nesse banco.

O banco de dados escolhido para armazenar essas informações é o MySQL. Essa escolha foi feita por se tratar de licença de software livre e de código aberto. Além disso, esse banco de dados é mundialmente utilizado, facilitando manutenções posteriores.

Abaixo o modelo do banco de dados.

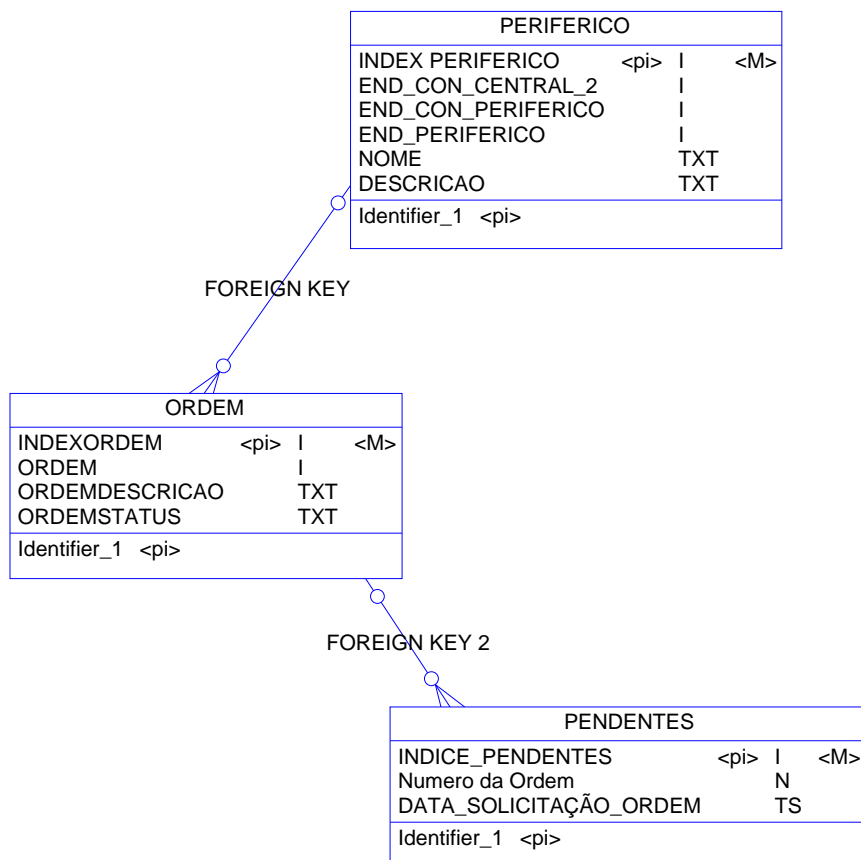
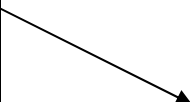


FIGURA 7.1 - Modelo do banco de dados

7.5. Estudo de caso

<i>Sistema de Automação Residencial</i>	
Caso de uso	Verificar estado
Ator	Usuário
Objetivo	Verificar estado dos periféricos do sistema
Breve descrição	O usuário vê como está o estado de cada periférico
Precondição	O usuário deve ter permissão

<i>Seqüência Típica de Eventos</i>			
Ação do autor		Resposta do Sistema	
1	O usuário seleciona o item de estado		
		2	O sistema mostra o estado do sistema



<i>Seqüência Alternativa de Eventos</i>	

<i>Sistema de Automação Residencial</i>	
Caso de uso	Especificar uma ordem
Ator	Usuário
Objetivo	O usuário envia uma ordem para ser executada
Breve descrição	O usuário ordenará uma função para o periférico
Precondição	O usuário deve ter permissão do sistema

<i>Seqüência Típica de Eventos</i>			
Ação do autor			Resposta do Sistema
1	O usuário seleciona o item de ordem		
			2 O sistema mostra os campos a serem inseridos: endereço do controlador do sistema, do controlador local dos periféricos, o endereço do periférico, nome e descrição.
3	O usuário digita os campos descritos e confirma a ordem (A1)		
			4 O sistema envia a ordem para o banco de dados.
5			
			6

<i>Seqüência Alternativa de Eventos</i>	
a1	Se o usuário pressionar o botão, cancela o programa e volta à página anterior

Sistema de Automação Residencial	
Caso de uso	Configurar sistema: inserir periférico
Ator	Usuário
Objetivo	O usuário configura o sistema
Breve descrição	Os usuários poderão inserir novos periféricos.
Precondição	O usuário deve ter permissão

Seqüência Típica de Eventos			
Ação do autor		Resposta do Sistema	
1	O usuário seleciona o item de configurar sistema.		
		2	O sistema mostra o botão “inserir”
3	O usuário clica o botão “inserir”		
		4	O sistema abre uma tela com vários campos para a configurar o novo hardware, tais como: o endereço do controlador do sistema, do controlador local do periférico, o endereço dos periféricoS e descrição para configurar.
5	O usuário configura os itens mostrados e clica “salvar”		
		6	O sistema mostra uma mensagem de confirmação. (A1)

Seqüência Alternativa de Eventos	
A1	Se existir algum conflito ou erro, o sistema mostrará um aviso, voltará a página e não salvará os dados.

Sistema de Automação Residencial	
Caso de uso	Configurar sistema: alterar informações do Periférico
Ator	Usuário
Objetivo	O usuário configura o sistema
Breve descrição	O usuário poderá alterar os periféricos, como por exemplo, a mudança de endereço dos periféricos já instalados.
Precondição	O usuário deve ter permissão

Seqüência Típica de Eventos			
Ação do autor		Resposta do Sistema	
1	O usuário seleciona o item configurar sistema.		
		2	O sistema mostrará o botão “alterar” dado do sistema
3	O usuário clica a opção		
		4	O sistema mostra todos os periféricos instalados e sua descrição.
5	O usuário seleciona o periférico		
		6	O sistema abre uma janela com todas as configurações desse periférico que pode ser alterado. E mostra dois botões: voltar e alterar.(A1)
7	O usuário altera as informações e solicita a alteração		
		8	O sistema confirma e volta à tela principal (A2)

Seqüência Alternativa de Eventos	
A1	Se o usuário selecionar, cancela o sistema e volta à página anterior
A2	Se houver algum conflito, o sistema mostra uma mensagem de erro e voltar à página anterior.

Sistema de Automação Residencial	
Caso de uso	Configurar sistema: inserir ordem
Ator	Usuário
Objetivo	O usuário configura o sistema
Breve descrição	O usuário terá a capacidade de inserir, deletar e alterar ordens do periférico.
Precondição	O usuário deve ter permissão

Seqüência Típica de Eventos			
Ação do autor		Resposta do Sistema	
1	O usuário seleciona o item configurar sistema.		
		2	O sistema mostrará o botão “modificação de ordem”.
3	O usuário clica a opção.		
		4	O sistema mostra todos os periféricos instalados e sua descrição, com a finalidade possibilitar ao usuário a escolha do periférico em que será feita a mudança na ordem.
5	O usuário seleciona o periférico e clica “continuar”		
		6	O sistema abre uma tela com todas as configurações de ordem do periférico sujeito a alteração. Mostra campos contendo o número da ordem e descrição que serão alterados. Mostra, ainda, três rádios: inserir, alterar e deletar, além de dois botões: voltar e continuar.(A1)
7	O usuário seleciona o rádio “inserir” e pressiona “continuar”.		
		8	O sistema confirma e volta à tela principal. (A2)

Seqüência Alternativa de Eventos	
A1	Se o usuário selecionar “voltar” o sistema, volta-se à página anterior
A2	Se houver algum conflito, o sistema mostra uma mensagem de erro e volta à página anterior.

Sistema de Automação Residencial	
Caso de uso	Configurar sistema: Alterar Ordem
Ator	Usuário
Objetivo	O usuário configura o sistema
Breve descrição	O usuário terá a capacidade de inserir, deletar e alterar ordens do periférico.
Precondição	O usuário deve ter permissão

Seqüência Típica de Eventos			
Ação do autor		Resposta do Sistema	
1	O usuário seleciona o item configurar sistema.		
		2	O sistema mostrará o botão ordem: “Modificação de Ordem”
3	O usuário clica a opção		
		4	O sistema mostra todos os periféricos instalados e sua descrição, com a finalidade possibilitar ao usuário a escolha do periférico em que será feita a mudança na ordem.
5	O usuário seleciona o periférico e clica “continuar”		
		6	O sistema abre uma tela com todas as configurações de ordem do periférico sujeito a alteração. Mostra campos contendo o número da ordem e descrição que serão alterados. Mostra, ainda, três rádios: inserir, alterar e deletar, além de dois botões: voltar e continuar.(A1)
7	O usuário seleciona o rádio “alterar” e clica “continuar”		
		8	O sistema confirma e volta à tela principal (A2)

Seqüência Alternativa de Eventos	
A1	Se o usuário selecionar “voltar” o sistema, volta-se à página anterior
A2	Se houver algum conflito, o sistema mostra uma mensagem de erro e volta à página anterior.

Sistema de Automação Residencial	
Caso de uso	Configurar sistema: Deletar Ordem
Ator	Usuário
Objetivo	O usuário configura o sistema
Breve descrição	O usuário terá a capacidade de inserir, deletar e alterar ordens do periférico
Precondição	O usuário deve ter permissão

Seqüência Típica de Eventos			
Ação do autor		Resposta do Sistema	
1	O usuário seleciona o item configurar sistema.		
		2	O sistema mostrará o botão “modificação de ordem”.
3	O usuário clica a opção		
		4	O sistema mostra todos os periféricos instalados e sua descrição, com a finalidade possibilitar ao usuário a escolha do periférico em que será feita a mudança na ordem.
5	O usuário seleciona o periférico e clica “continuar”.		
		6	O sistema abre uma tela com todas as configurações de ordem do periférico sujeito a alteração. Mostra campos contendo o número da ordem e descrição que serão alterados. Mostra, ainda, três rádios: inserir, alterar e deletar, além de dois botões: voltar e continuar.(A1)
7	O usuário seleciona o botão “deletar”		
		8	O sistema confirma e volta à tela principal (A2)

Seqüência Alternativa de Eventos	
A1	Se o usuário selecionar “voltar” o sistema, voltar à página anterior
A2	Se houver algum conflito, o sistema mostra uma mensagem de erro e volta à página anterior.

Sistema de Automação Residencial	
Caso de uso	Enviar ordem
Ator	Usuário
Objetivo	O usuário envia uma ordem ao sistema
Breve descrição	O usuário terá a capacidade de enviar uma ordem ao periférico
Precondição	O usuário deve ter permissão

Seqüência Típica de Eventos			
Ação do autor		Resposta do Sistema	
1	O usuário seleciona o item ordem do sistema.		
		2	O sistema mostrará uma tela que deverá ser preenchida pelo usuário. Os dados são: índice periférico, endereço controlador central, endereço controlador periférico, endereço periférico e número da ordem. Serão mostrados mais três botões: voltar (A1), enviar, pesquisar (A2).
3	O usuário clica a opção “enviar”		
		4	Aparece uma tela de confirmação da solicitação e um botão “retornar”.
5	O usuário clica o botão “retornar”		
		6	O Sistema volta à tela principal

Seqüência Alternativa de Eventos	
A1	Se o usuário selecionar “voltar” o sistema, volta à página anterior.
A2	Se o usuário clicar em “pesquisar”, o sistema abre uma tela de pesquisa (ver estudo de caso: pesquisa).

Sistema de Automação Residencial	
Caso de uso	Pesquisa
Ator	Usuário
Objetivo	O usuário faz uma pesquisa
Breve descrição	O usuário terá a capacidade de pesquisar um determinado periférico e suas ordens.
Precondição	O usuário deve ter permissão

Seqüência Típica de Eventos			
Ação do autor		Resposta do Sistema	
1	O usuário seleciona o item pesquisar.		
		2	O sistema mostrará uma tela com todos os periféricos instalados e um <i>input</i> com o índice do periférico escolhido que deverá ser preenchido pelo usuário. Serão mostrados mais dois botões: pesquisar e voltar (A1)
3	O usuário clica a opção “pesquisar”		
		4	Aparece uma tela de com todas as ordens possíveis desse periférico.

```

graph LR
    1[1: O usuário seleciona o item pesquisar.] --> 2[2: O sistema mostrará uma tela com todos os periféricos instalados e um input com o índice do periférico escolhido que deverá ser preenchido pelo usuário. Serão mostrados mais dois botões: pesquisar e voltar (A1)]
    3[3: O usuário clica a opção "pesquisar"] --> 4[4: Aparece uma tela de com todas as ordens possíveis desse periférico.]
  
```

Seqüência Alternativa de Eventos	
A1	Se o usuário selecionar “voltar” o sistema, volta à página anterior.

<i>Sistema de Automação Residencial</i>	
Caso de uso	Pendência do sistema
Ator	Usuário
Objetivo	O usuário faz uma pesquisa das pendências do sistema
Breve descrição	O usuário terá a capacidade de pesquisar as pendências do sistema.
Precondição	O usuário deve ter permissão

<i>Seqüência Típica de Eventos</i>			
Ação do autor		Resposta do Sistema	
1	O usuário seleciona o item “pendência” do sistema.		
		2	O sistema mostrará uma tela com todas as pendências do sistema. Serão mostradas as características do sistema e o botão “voltar”.
3	O usuário clica a opção “voltar”		
		4	O sistema volta à tela de início

<i>Seqüência Alternativa de Eventos</i>	

7.6 Tela do programa do usuário

Enviar Ordem ao sistema

Clique para digitar o index do periférico 1

Ou clique para digitar o endereço completo

Endereço Controlador Central 0

Endereço Controlador Periférico 0

Endereço Periférico 1

Digite o código da ordem 1

data 23/06/2004

hora 01:07

enviar voltar pesquisar

FIGURA 7.2 – Pagina de envio de ordem ao sistema

Inserindo novo hardware

Índice periférico	0
Nome do Periférico	
Endereço Controlador Central	0
Endereço Controlador Periférico	0
Endereço Periférico	0
Descrição	

Voltar Inserir

FIGURA 7.3 – Inserindo novo hardware.

Página de configuração

Dados do periférico

<i>índice</i>	<i>Endereço Controlador Central</i>	<i>Endereço Controlador Periférico</i>	<i>Endereço Periférico</i>	<i>Nome</i>	<i>Descrição</i>
2	1	2	3	Porta	Porta da sala

Digite o de index 0

Digite o código da ordem

Digite o texto que descreve essa ordem

☒ Inserir ordem

[Voltar](#)

[Continuar](#)

FIGURA 7.4 Inserindo ordens ao periférico

Status do sistema

<i>índice</i>	<i>Endereço Controlador Central</i>	<i>Endereço Controlador Periférico</i>	<i>Endereço Periférico</i>	<i>Nome</i>	<i>Descrição da Ordem</i>	<i>Código ordem</i>
3	1	0	1	Janela teste	Não definido	0

[Voltar](#)

FIGURA 7.5 Verificando estado do periférico

Pesquisa dos hardware instalados

índice	Endereço Controlador Central	Endereço Controlador Periférico	Endereço Periférico	Nome	Descrição
1	1	0	1	Janela teste	Janela de teste

Digite o index do periférico 1

[Voltar](#) [Pesquisar](#)

FIGURA 7.6 Pesquisa de *hardwares* instalados

Pesquisa dos hardware instalados

Especificações de ordem

Código ordem	Descrição ordem
1	abre
2	fecha
0	Não definido

Digite o index do periférico 1

[Voltar](#) [Pesquisar](#)

FIGURA 7.7 Pesquisa de ordem do *hardware*

Capítulo 8. PERIFÉRICO

O periférico criado tem a finalidade de ligar um pequeno motor que funciona no sentido horário e anti-horário dependendo do estado dos Flip-Flop`s. Esses Flip-Flop`s do tipo JK têm suas entradas ligadas à tensão de alimentação do circuito. Na figura dezoito, visualiza-se como foi feita a implementação destes Flip-Flop`s. O estado desses Flip-Flop`s pode ser modificado por meio de ordens recebidas do microcontrolador ou alterado diretamente pelo usuário no hardware.

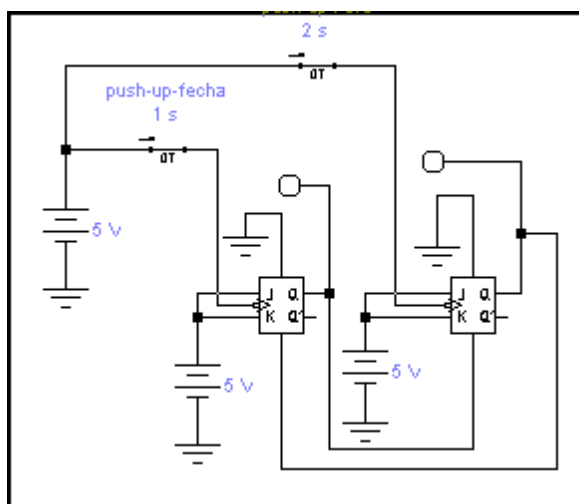


FIGURA 8.1 - Memória do sistema

Esse projeto está utilizando um sensor magnético que é composto de duas partes. A primeira parte é o imã; a segunda é uma resistência que é sensível às ondas eletromagnéticas emitidas pelo imã. Quando o imã se aproxima da resistência, isso faz com que a haja uma resistência baixa e, ao afastá-lo, sua resistência é alta, da ordem de *megaohms*.

A figura 8.2 é composta por sensores e portas lógicas que funcionam como controle do relé. Nessa figura, são vistos dois circuitos compostos de duas portas lógicas *and* e dois inversores de sinais. Cada circuito tem a possibilidade de ter dois tipos de sensores e uma entrada que simbolizam a ordem a ser executada. Caso os sensores enviem o nível lógico “um” (cinco volts) para o inversor, isso muda o estado final do circuito. Nesse caso, o relé é desligado.

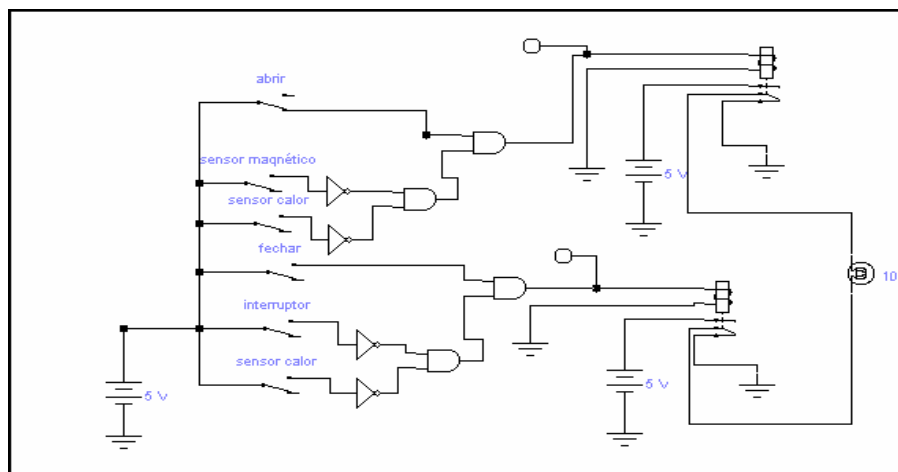


FIGURA 1.2 - Controle que liga e desliga o motor e sensores de controle

A figura 8.3 tem a finalidade de gerar um pulso na entrada de *clock* do Flip-flop JK. Esse circuito tem três finalidades: a primeira é de *reset* dos Flip-flop quando o sensor for ativado; a segunda finalidade ocorre quando o microcontrolador ou usuário envia alguma ordem ao periférico; a terceira é impedir que haja mudança do estado dos Flip-flop's por meio da entrada analógica (comando do usuário) e digital (comando do microcontrolador) se o sensor magnético estiver ativado.

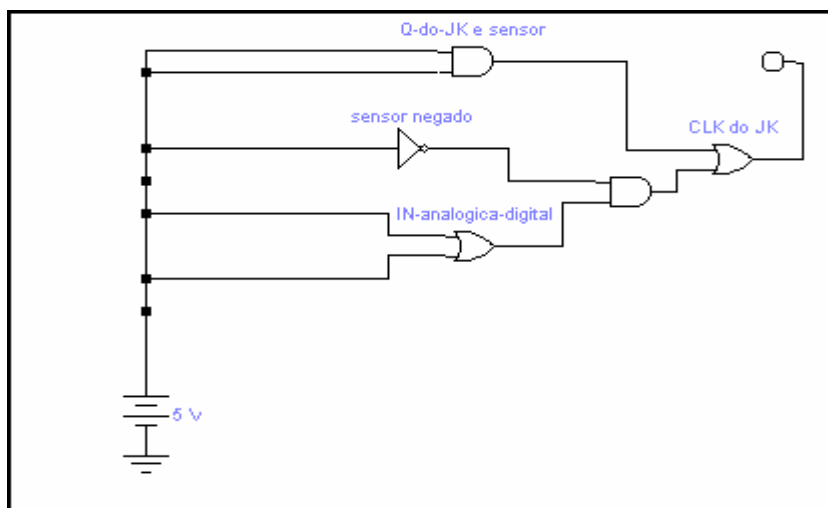


FIGURA 8.3 - Controle do sistema

A figura 8.4 a ser exibida é o circuito completo que representa a união dos blocos citados anteriormente.

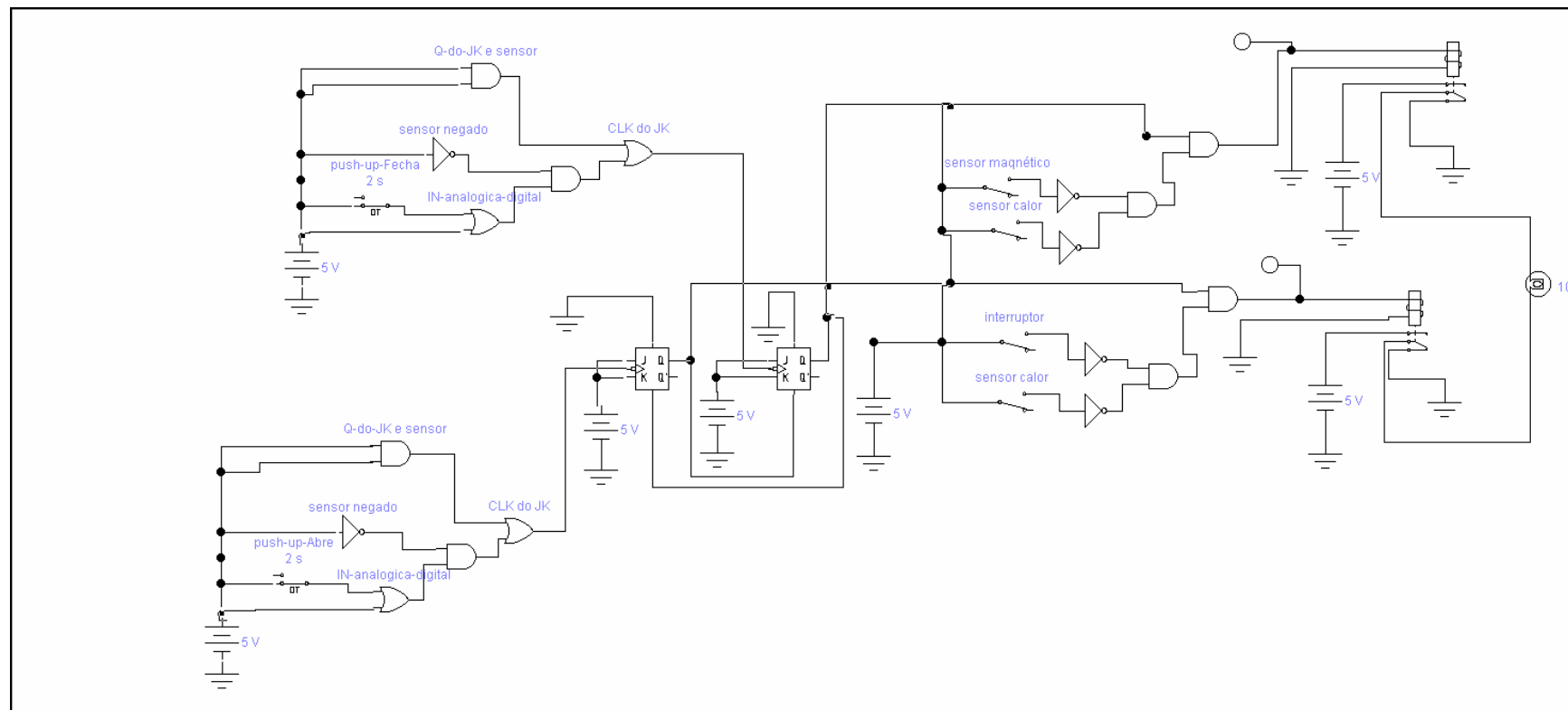


FIGURA 8.4 - Circuito Completo

Capítulo 9. PRÉ-REQUISITOS DO SISTEMA

Como foi visto nessa monografia foi utilizado vários programas que podem se encontrados no *sítes* já mencionado. Para o funcionamento do Java é preciso instalar o j2sdk que, hoje, se encontra na versão 1.4.2. Segundo a manual do fabricante para o funcionamento é preciso um Pentium 166MHz ou mais veloz processador de no mínimo de 32 megabytes RAM para funcionar o aplicativo gráfico. No mínimo de 48 *megabytes* de RAM é recomendada para o *applets* que funciona do *browser* que usa o *plug-in* do Java *Plug-in*. As faltas desses recursos podem causar dano na performance ou inviabilizar o funcionamento.

Além do j2sdk é necessário a servidor TOMCAT que está na sua versão 5. Para o seu funcionamento é necessário que o j2sdk esteja instalado na máquina antes de instalá-lo.

Na atual implementação faz-se necessário uma porta comunicação serial que normalmente é chamado de com. Esse é o meio de comunicação entre o servidor e a UCP.

Capítulo 10. CONCLUSÃO

Nesse projeto final, foi apresentada uma sugestão de implementação de um sistema de automação residencial com o intuito de ajudar os deficientes paraplégicos, principal meta desse projeto.

Para se atingir a meta, foram aplicados conceitos como o sistema distribuído, redes de computadores, lógica digital, lógica de programação além da utilização de linguagem de programação Java e C.

Todos esses elementos, considerados na sua totalidade, culminaram na criação de um periférico e de programas mais complexos, descritos neste projeto com detalhamento técnico, que, juntos, representam uma solução para simulação de um sistema de automação residencial voltado para o deficiente paraplégico. Com sua implementação, o usuário terá conhecimento do estado de sua residência, ou seja, verificará se as portas e as janelas estão abertas ou fechadas e executará o ato de abrir ou fechá-las, sem necessidade de se locomover. Assim, nesse projeto foram implementados os programas que interagirão com o usuário, sendo eles o software controlador que tem a finalidade de unir o *software* do usuário com o sistema de automação, um protocolo de comunicação entre o computador e o 8051, o software do microcontrolador 8051. Um periférico foi implementado para ligar um motor no sentido horário e anti-horário. Para isso, foram utilizados sensores, portas lógicas e relés.

Os testes de software e de hardware atenderam às expectativas do projeto, pois, tanto o *software* do usuário quanto o *software* controlador, o protocolo e o periférico funcionaram perfeitamente em conjunto com o hardware proposto, criando, assim, um conjunto que simula um sistema de automação residencial.

Embora primeiramente tenha um enfoque de simulação, este projeto poderá tornar-se realidade, na medida em que for considerada a funcionalidade da ferramenta e incorporada à realidade das pessoas.

Como sugestão para futuros trabalhos, em continuidade a esse projeto, podem ser citados: a implementação da Unidade Central de Controle (UCC), o aperfeiçoamento do protocolo de comunicação, a criação de uma estrutura de redes para transmitir os dados dos periféricos das suas respectivas unidades e o

aperfeiçoamento da segurança de todo o sistema, temas que não fizeram parte do escopo desse trabalho. A também a possibilidade de utilizar o celular para interagir com o usuário por meio da tecnologia MIDP que compõe alguns celulares.

Bibliografia

- [1] HORSTMANN, Cay S. E Cornell, Gary. Core Java 2 Volume 2 Recursos Avançados. São Paulo: Makron Books Ltda, 2001
- [2] IRWIN, J. David. Análise de circuitos em engenharia. 4.ed. São Paulo: Makron Books, 2000.
- [3] GIMENEZ, Salvador P. Microcontroladores 8051. São Paulo: Makron Books, 2002.
- [4] MEDEIROS, João Bosco. Redação Científica: A prática de fichamento, resumos, resenhas. 6 ed. São Paulo: Atlas, 2004.
- [5] PAZOS, Fernando. Automação de sistema & Robótica. Rio de Janeiro: Axcel Books do Brasil editora LTDA, 2002.
- [6] SOARES, Luiz Fernando Gomes. Redes de Computadores: das LANs, MANs e WANs às Redes ATM .2 ed. Rio de Janeiro: Campus, 1995.
- [7] TANENBAUM, Andrews S. Organização estruturada de computadores. 3. ed. Rio de Janeiro: LTC, 1999.
- [8] TANENBAUM, Andrews S. Redes de computadores. 3. ed. Rio de Janeiro: Campus, 1997.
- [9] THEODORE F. Bogard, JR. Dispositivos e circuitos eletrônicos volume 1. São Paulo: Makron Books, 2001.
- [10] THEODORE F. Bogard, JR. Dispositivos e circuitos eletrônicos volume 2. São Paulo: Makron Books, 2001.
- [11] <http://www.linhadecodigo.com.br/desenvolvimento/cpp.asp>
- [12] <http://www.mysql.com/>
- [13] <http://sdcc.sourceforge.net/>
- [14] PHILIPS. Datasheet **80C552/83C552** Single-chip 8-bit microcontroller. Estados Unidos da America: Philips, 1998